

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

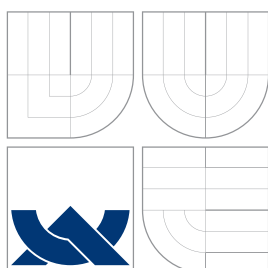
POUŽITÍ MODERNÍCH KOMUNIKAČNÍCH FORMÁTŮ PRO IMPLEMENTACI INFORMAČNÍHO SYSTÉMU

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

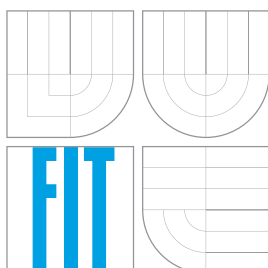
AUTOR PRÁCE
AUTHOR

MILAN PÁLA

BRNO 2010



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

POUŽITÍ MODERNÍCH KOMUNIKAČNÍCH FORMÁTŮ PRO IMPLEMENTACI INFORMAČNÍHO SYSTÉMU

USING OF MODERN COMMUNICATION FORMATS FOR IS IMPLEMENTATION

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

MILAN PÁLA

VEDOUCÍ PRÁCE

SUPERVISOR

prof. Ing. TOMÁŠ HRUŠKA, CSc.

BRNO 2010

Abstrakt

Tato práce se zabývá analýzou požadavků a tvorbou prototypu informačního systému hasičské sportovní ligy. Ten vychází z dosavadní verze postavené na obecném publikačním systému, jehož možnosti nedostačují požadavkům na moderní informační systém. Součástí nové verze systému jsou grafy, na kterých byl nastíněn současný stav moderních zobrazovacích prostředků a pomocí technologie Silverlight vytvořen modul, který grafy pro aplikaci vykresluje.

Abstract

This thesis is about analysis of requirements and creation of a prototype of information system of firesport league. System starts from existing version which is based on general content management system which does not include properties to make a modern IS. Charts represent the part of a new version of the system. By the charts was outlined current state of modern communication formats. Through the Silverlight was created a module which is able to draw the charts for the application.

Klíčová slova

grafy, Silverlight, Silverlight Toolkit, PHP, MySQL, Nette Framework, požární sport, informační systém

Keywords

chart, Silverlight, Silverlight Toolkit, PHP, MySQL, Nette Framework, firesport, information system

Citace

Milan Pála: Použití moderních komunikačních formátů pro implementaci informačního systému, bakalářská práce, Brno, FIT VUT v Brně, 2010

Použití moderních komunikačních formátů pro implementaci informačního systému

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana profesora Hrušky.

.....

Milan Pála
18. května 2010

Poděkování

Rád bych poděkoval panu profesorovi Hruškovi, za možnost realizovat návrh systému, jehož použití je mi blízké a práce na něm mě bavila a obohatila.

© Milan Pála, 2010.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1	Úvod	3
2	Systém soutěží Krušnohorské Ligy a vyhodnocování hasičských soutěží	4
3	Analýza moderních komunikačních prostředků ve webovém prostředí	5
3.1	Klasické prostředky	5
3.2	Adobe Flash	5
3.3	Microsoft Silverlight	6
3.4	HTML 5	6
4	Analýza požadavků na informační systém	7
4.1	Nevýhoda současného řešení	7
4.2	Návrh na zlepšení	7
4.2.1	Evidence družstev, sborů, terčů	8
4.2.2	Možnost přihlašování na závody	8
4.2.3	Hodnocení pořádání závodu	10
4.2.4	Diskuze	10
4.2.5	Vkládání výsledků	10
4.2.6	Uživatelé a oprávnění	11
4.2.7	Další možnosti přihlášení	11
4.2.8	Správa systému	12
4.2.9	Automatická kontrola obsahu	13
4.2.10	Statistiky	13
5	Analýza požadavků na zpracování grafů	14
5.1	Určení dat vhodných pro reprezentaci pomocí grafů	14
5.1.1	Koláčový graf	14
5.1.2	Spojnicový graf	14
5.1.3	Pruhový graf	16
5.2	Formát dat	16
5.3	Integrace do systému	17
6	Použité technologie	18
6.1	Pro systém	18
6.1.1	PHP	18
6.1.2	JSON	18
6.1.3	AJAX	18
6.1.4	Nette framework	19

6.1.5	dibi framework	20
6.2	Pro grafy	20
6.2.1	XAML	20
6.2.2	C#	20
6.2.3	Silverlight a Silverlight Toolkit	20
7	Implementace systému	22
7.1	Struktura aplikace	22
7.2	Využití komponent pro vložení grafu do stránky	22
7.3	Správa systému	23
7.4	Možnosti diskuzí a souvisejících položek	23
7.5	Fotogalerie	24
7.6	Statistiky	25
8	Implementace grafů	26
8.1	Úvod	26
8.2	Návrh implementace	26
8.3	Moje implementace	26
8.3.1	Stažení dat	27
8.3.2	Zpracování dat	27
9	Zhodnocení výsledků	29
9.1	Přínos systému pro Krušnohorskou ligu	29
9.2	Přínos systému pro autora	29
9.3	Návrh dalších prací na systému	29
A	Obsah CD	33

Kapitola 1

Úvod

V dnešní době, kdy se internet stal fenoménem a práci s ním zvládají jak mladí, tak staří, není jednodušší a levnější způsob prezentace, než se zviditelnit na síti sítí. Firma, podnikatel či organizace, která nemá na internetu svou, aspoň skromnou, vizitku, má velkou konkurenční nevýhodu. Přitom možností, jak se na internetu zviditelnit, je mnoho. Od prostých několika stránkových prezentací, po rozsáhlé informační systémy.

Oboje je dnes člověk schopen zvládnout bez větších infromatických znalostí. Uživatelsky přívětivá rozraní publikačních systémů nabízí mnoho poskytovatelů internetového prostoru. Kolikrát i zadarmo, pouze za cenu domény, avšak i bez ní. Složitějším případem je využití veřejně dostupného publikačního systému. I tak je prvotní nastavení otázkou pročtení nápovědy, či literatury. Takovéto prezentace jistě postačují mnoha subjektům po dlouhou dobu.

Problém nastane, pokud požadavky na systém překročí možnosti obecného publikačního systému. Požadavky nemusí být nijak vysoké. Stačí banalita, která je ovšem specifická pro cílovou skupinu. Potom je nejlepší vytvořit redakční systém na míru, který dobře pokryje potřeby zákazníka. Nejde při tom jen o požadavky na prezentování dat, ale také na snadnou obsluhu systému a další práci s daty, jako může být tvorba tiskových sestav nebo generování statistik. Přesně o takovémto systému je tato technická zpráva.

Kapitola 2

System soutěží Krušnohorské Ligy a vyhodnocování hasičských soutěží

Tento informační systém je vyvíjen pro potřeby Krušnohorské ligy.

Krušnohorská liga je seriál soutěží v požárním útoku mužů, žen a nově i veteránů. V okolí Krušných hor působí již od roku 1997, přičemž postupem času se rozšířila po celém Ústeckém kraji. Někdy se Ligy zúčastní i družstva ze vzdálenějších okresů, jako jsou okresy ze Západočeského, Středočeského a Libereckého kraje. Výjimečně se zapojí i družstva profesionálních hasičů či jednotek z Německa. Ze všech disciplín požárních soutěží se závodí pouze v disciplíně požární útok. [14]

Při něm se jedno sedmičlenné družstvo snaží trefit proudem vody dva terče v co nejrychlejším čase, přičemž se jako výsledný čas počítá čas horší. Z informativních důvodů se zaznamenává také čas rychleji sestřeleného terče. Během jednoho soutěžního dne zápolí mnoho družstev různých sborů, která se do závodu zaregistrovala prostřednictvím internetových stránek Ligy. Od sezóny 2009 přibyla k dvěma stálým kategoriím mužů a žen ještě doplňková kategorie veteránů, která je nabízena pouze jako alternativa pro pořadatele. Zda bude soutěž pořádána i pro tuto kategorii, oznámí pořadatel před zahájením sezóny a její výsledky se evidují obdobným způsobem jako ostatních dvou kategorií.

Pořadatel zároveň volí, na jaké druhy terčů soutěž uskuteční – sklopné nebo plnicí. Časy družstev na oba druhy jsou vzhledem k charakteru terčů odlišné. Proto je nutné vést dvojí statistiky výsledných časů.

V průběhu jednoho ročníku družstva sbírají body podle dosaženého umístění na jednotlivých závodech. Pro každou kategorii je stanoveno jiné bodové ohodnocení za dosažené pořadí. Na posledním závodě probíhá vyhodnocení celého ročníku a podle součtu dosažených bodů ze všech závodů se určí pořadí jednotlivých družstev ve své kategorii.

Výsledky ze všech soutěží jsou zveřejněny na internetových stránkách Ligy. Stejně tak se udržuje i tabulka průběžného stavu celkových bodů. Ze závodů jsou vedeny základní statistiky, jako jsou nejrychlejší dosažené časy a vítězové jednotlivých ročníků.

Kapitola 3

Analýza moderních komunikačních prostředků ve webovém prostředí

3.1 Klasické prostředky

Samotný jazyk HTML v současné verzi nenabízí žádné možnosti pro pokročilejší grafická zobrazení. Je to dáno především původem jazyka. Jistou možnost nabízí stylovací jazyk CSS. Konkrétně několik studií grafů tvořených pouze pomocí HTML a CSS již bylo zveřejněno. [5] Mnohdy je potřeba takového řešení doplnit dodatečnými obrázky. A ani v tomto případě nemusí být řešení ideální vzhledem k různým implementacím standardů v prohlížečích.

Další volbou může být zobrazení grafů pomocí obrázků. Serverové skriptovací jazyky nabízí podporu pro práci s grafickými knihovnami¹ a je tak nasnadě vytvořit modul, který graf vykreslí na serveru a vygenerovaný obrázek pošle klientovi do prohlížeče. Výhodou generování obrázků je možnost jejich cachování jak na serveru, tak u klienta v dočasných souborech prohlížeče. Kontrolní otisk, pomocí kterého je obrázek cachovacími subsystémy identifikován, je možné generovat ze vstupních dat. Obrázek grafu zůstane platný, dokud se data pro graf nezmění. Nevýhodou může být nedostupnost vyhlazovacích algoritmů v grafických knihovnách. Grafy často disponují množstvím šikmých čar, křivek a textů. Hranaté okraje takových elementů mohou působit nevzhledně. Obrázky také neumožňují uživatelskou interakci – zvýraznění aktuální datové řady, detail části grafu apod.

Všechny výše zmíněné nevýhody stály za myšlenkou vytvořit technologii, která by umožnila využít potenciál klientských stanic, co se možnosti grafického zobrazení týče. V současné době jsou na trhu tři technologie², umožňující tvoření graficky pokročilých uživatelských rozhraní.

3.2 Adobe Flash

Vzhledem k nejvyššímu rozšíření³ se jeví použití technologie Flash jako nejvýhodnější. Bohužel pro vývoj aplikací je nutné využít komerční prostředí. Tato skutečnost v tomto případě

¹PHP: *Hypertext Preprocessor* [online]. 2010 [cit. 2010-05-08]. PHP: Image Processing and Generation - Manual. Dostupné z WWW: <<http://cz.php.net/manual/en/refs.utilspec.image.php>>.

²Adobe Flash, Microsoft Silverlight a JAVA od firmy Oracle. Tuto technologii jsem neuvažoval.

³Viz [26]. Uváděné statistiky se zaměřují především na trh v USA, pro názornost rozšíření technologií to ovšem nevádí

znemožňuje další použití technologie.

3.3 Microsoft Silverlight

Poměrně mladá technologie nabízí podobné vlastnosti pro tvorbu moderních internetových aplikací jako Adobe Flash. Kromě samotného jádra je nabízen Silverlight Toolkit [16], který nabízí mnoho vizuálních i nevizuálních komponent pro snadnější a přímochařejší vývoj. Součástí je také komponenta Charting, poskytující rozhraní pro tvorbu několika typů grafů.

Pro vývoj aplikací se používá integrované prostředí Microsoft Visual Studio. Kompatibilita s rozhraním Silverlight a Silverlight Toolkit je zaručena buď komplexním balíkem aplikací Microsoft Web Platform nebo samostatným doplňkem Silverlight 3 Tools for Visual Studio⁴. Všechny potřebné aplikace jsou distribuované zdarma, komerční aplikace Microsoft Visual Studio je pro studenty k dispozici zdarma prostřednictvím programu MSDN AA, popřípadě zdarma prostřednictvím verze Express.

3.4 HTML 5

Součástí nové verze jazyka HTML se chystá nový element canvas. [11] Pomocí jazyka JavaScript je možné do vytvořeného kreslicího plátna malovat a vytvářet pokročilé grafické výtvořiny včetně 3D prostředí. Výhodou použití tohoto řešení je podpora v každém prohlížeči implementující HTML 5 bez nutnosti dodatečné instalace zásuvných modulů třetích stran. Oproti použití technologií Flash nebo Silverlight je výhodou menší datová náročnost kódu nutného pro kreslení a zároveň odpadá nutnost používat speciální vývojová prostředí nebo doplňky nutné pro kompilaci zdrojových kódů. V současné době není standard hotov a tudíž ani podpora ve webových prohlížečích není ideální. Do budoucna lze předpokládat, že nový standard HTML 5 Flash a Silverlight do jisté míry vytlačí.⁵

⁴Home : The Official Microsoft Silverlight Site [online]. 2010 [cit. 2010-05-08]. Get Started : The Official Microsoft Silverlight Site. Dostupné z WWW: <<http://www.silverlight.net/getstarted/>>.

⁵O tomto trendu svědčí i snaha serveru Youtube.com využít místo Flashového přehrávače element video z HTML 5. [19] Další informace o pokračující kodifikaci normy v češtině v [6]

Kapitola 4

Analýza požadavků na informační systém

Hlavní úlohou informačních systémů by mělo být ulehčení rutinní práce uživatelů. Správný systém by se měl přizpůsobit potřebám uživatele natolik, aby s ním pracoval rád, obzvláště pokud je veškerý obsah generovaný uživatelem. Pokud by práce se systémem nebyla intuitivní, uživatelé by práce se systémem odrazovala a obsah by přidávali neradi. Takový systém je zralý k přepracování, či vytvoření nového systému.

4.1 Nevýhoda současného řešení

Internetové stránky představují výborný způsob k prezentaci informací. Při vhodném využití hypertextových odkazů se snadno uvede spoustu informací do souvislosti s jinými a čtenář snadno získá ucelený přehled znalostí.

Současný systém, vzhledem ke svému stáří a použití obecného publikačního systému, nebyl schopen vhodně využít všechny nabízené aspekty moderního internetu. Výsledky jsou evidovány bez dalších souvislostí. Například bez příslušnosti jednotlivých výsledků k družstvům nebo různých statistik výsledků jak z jednotlivých ročníků, tak celkově. Jediná vedená statistika nejlepších časů vykazuje chyby v zobrazovaných údajích. Očividně vypisuje konstanty, které se měly interpretovat jako zvláštní případy (diskvalifikace apod.), jako platné výsledky.

Systém zároveň nabízí pouze ruční možnost spouštění rezervací na soutěže. Tato vlastnost byla jedním z redaktorů systému při analyzování požadavků hodnocena záporně, protože musí často provádět kontrolu práce pořadatelů závodů a možnost registrace se otvírá s mírným zpožděním.

4.2 Návrh na zlepšení

Aby bylo možné provést všechny změny, bylo nutné navrhnout systém zcela od počátku s ohledem na cílové požadavky a zaměření systému.

Hlavním požadavkem rady KL bylo, aby zůstala zachována současná funkčnost. A to především prezentace výsledků ze soutěží, zveřejňování článků, možnosti diskuzí a zobrazování fotografií.

Nejdříve proběhla analýza všech požadavků a podle nich návrh ER diagramu. Současný stav byl rozšířen o tyto vlastnosti:

- evidence jednotlivých družstev, které se závodů kdy účastnily,
- evidence všech použitých terčů,
- evidence sborů, které se účastnily závodů, vlastní používané terče nebo organizovaly závody,
- možnost nastavení automatického spuštění přihlašování družstev na závody,
- možnost hodnotit pořádání závodu,
- rozšířená možnost diskuzí k závodům, sborům, družstvům, terčům a článkům,
- možnost vkládání videí z veřejně dostupných serverů nabízejících streamovaný obsah,
- vedení několika nových sportovních a organizačních statistik Ligy.

4.2.1 Evidence družstev, sborů, terčů

V současném systému není návrh databáze pro ukládání výsledků ideální. Je určen především pro jednoduchý výpis výsledků k jednotlivým závodům. Vhodným návrhem nové struktury půjde snadno evidovat a získávat všechna potřebná data.

Při návrhu ER diagramu byly použity entity závody, výsledky, družstva. S nimi souvisí další entity, jako jsou sbory, kategorie a místa. Výsledky jsou evidovány k závodu. Výsledku dosáhlo na závodě družstvo, které náleží sboru a na závodech soutěží v jedné kategorii. Závody pořádá hasičský sbor, ale ne vždy ve své vlastní obci, může závod uspořádat i jinde. S tím souvisí evidence míst, které se vztahují buď ke sborům, nebo k místům pořádání závodů.

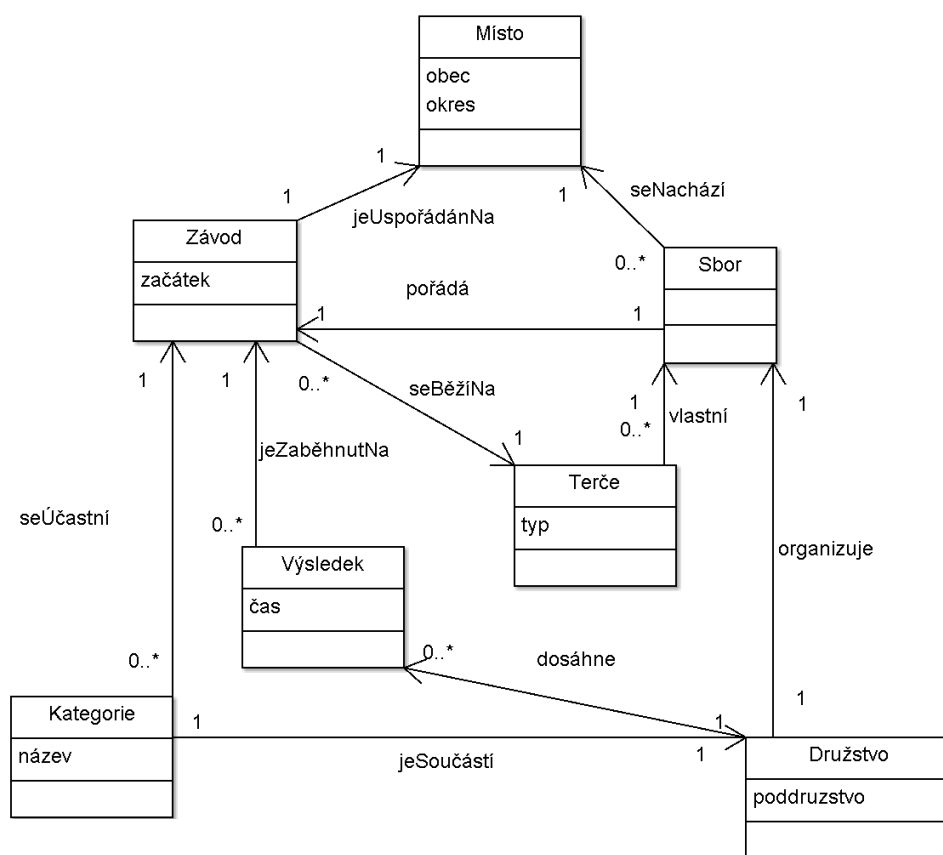
Nově bude systém umět evidovat i informace o používaných terčích. Nyní se ukládá název sboru, kterému terče patří a jejich typ. Navrhl jsem tedy terče modelovat jako samostatnou entitu. Díky tomu půjde snadno vést statistiky pro každý druh terčů zvlášť.

4.2.2 Možnost přihlašování na závody

Přihlášení družstev na závody je podle pravidel podmíněno registrací na internetových stránkách Ligy. Současný systém není na tuto možnost připraven a pořadatel vždy v diskuzi zakládá nové téma, do kterého zástupci jednotlivých týmů potvrzují svou účast a rezervují si startovní místo. Často se stává, že pořadatelé téma nezaloží a ostatní správci musí zakládat téma za ně. Zároveň se již příspěvky s přepsanými přihláškami odstraňují. Tím se ztrácí historie a nedá se dohledat, kdo a kdy startovní pořadí zapsal.

Z těchto důvodů jsem připravil automatickou registraci. Na začátku sezóny při vytváření závodů se ke každému závodu uvede, které kategorie se závodu budou účastnit, a k nim se uvede počet předpokládaných startovních míst. Předposlední pondělí před závodem se na stránce každého závodu otevře možnost přihlášení družstva do volné startovní pozice. Tento termín jsem zvolil, aby na přihlášení bylo aspoň 10 dní a vzhledem k pořádání závodů v sobotu i neděli se otevřelo přihlašování na oba závody současně.

Z vlastní zkušenosti znám případy, kdy týmy nemají svého zástupce, který by je přihlásil, a přihlášku za ně podává člen jiného týmu. Z tohoto důvodu bude moc přihlásit kterýkoli přihlášený uživatel, kterékoli družstvo chce. Odebrat přihlášku ale bude moc už kterýkoli člen sboru dotčeného družstva. Správce systému a pořadatel budou mít ještě více volnou



Obrázek 4.1: ER diagram části související se správou závodů a výsledků.

ruku, a to v podobě plného přístupu ke všem přihláškám, kde mohou měnit pořadí přihlášených týmů nebo rušit kterékoli přihlášky. Systém je založen na loajalitě uživatelů uvést při registraci správně příslušnost ke sboru.

Navrhované řešení má minimálně jednu nevýhodu – je nutné uvést předpokládaný počet startujících v každé kategorii. Některé týmy chtějí soutěžit jako poslední, což by při postupném přidávání nových startovních pozic podle zapsaných týmů nešlo. Možností by bylo evidovat příznak, že požadované startovní pořadí je poslední, tím by se ale ztížil současný návrh implementace nadmíru výslednému řešení. Vzhledem k relativně stálému počtu přihlašovaných týmů bylo zvoleno právě jednodušší řešení v podobě pevně stanoveného počtu pozic.

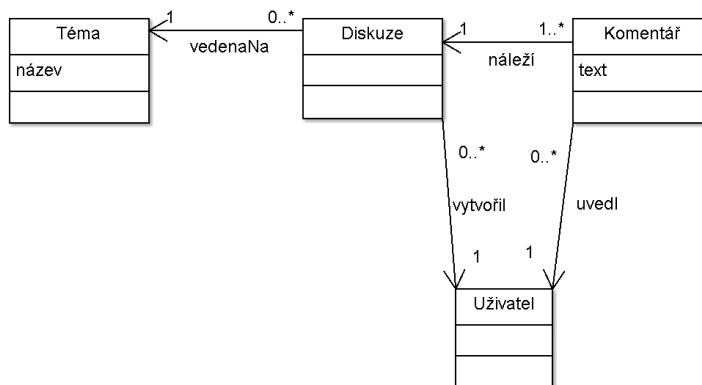
4.2.3 Hodnocení pořádání závodu

Aby bylo možné pořadatelům veřejně sdělit, jak se pořádání závodu podařilo, připravil jsem možnost ohodnotit závod udělením jedné až pěti hvězdiček. Ohodnotit lze všechny uspořádané závody.

4.2.4 Diskuze

Nyní existovala diskuze jako taková, ve které byly předvolená témata, do kterých uživatelé pokládali své dotazy. Pouze ke článkům bylo možné přidávat komentáře, nicméně nezávislé na diskuzi.

Možnosti diskuzí jsem tedy rozšířil o možnost komentování závodů, družstev, sborů a terčů. Včetně článků je diskuze přístupná jak z hlavní stránky diskuzí, tak jednotlivé diskuze jsou přístupné u související položky. Stávající možnosti, jako je zamknutí diskuze, zůstaly zachovány.



Obrázek 4.2: ER diagram diskuzí.

4.2.5 Vkládání výsledků

Zřejmě nejdůležitější a nejčastější prací se systémem ze strany správců systému bude evidence výsledků ze závodů. Uživatel bude zadávat velké množství dat, která bude s největší pravděpodobností opisovat z vytištěných výsledků ze závodu. Zadávání by tedy mělo probíhat plynule a uživatel by měl používat pouze klávesnici.

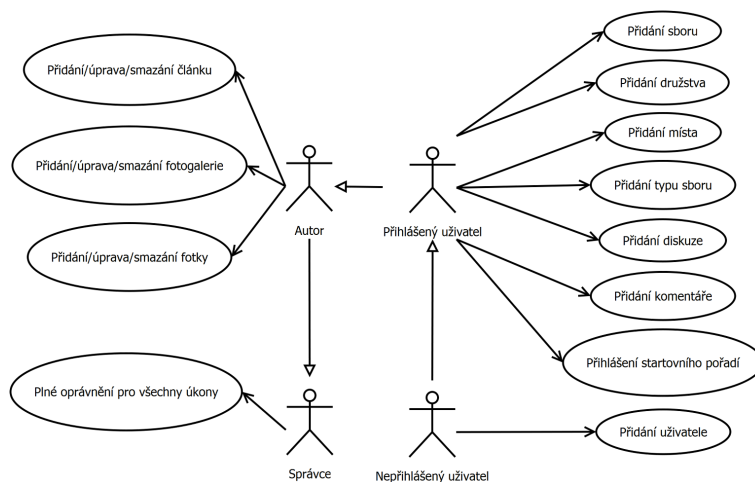
Uživatel uvádí družstvo, výsledky lepšího a horšího terče respektive informaci o neplatném pokusu. Ze závodu je evidováno dosažené pořadí a body. Oba dva údaje lze určit i automaticky a ušetří tak uživateli zadávání nadbytečných informací. Po zadání výsledků lze tedy nechat systém dopočítat podle dosažených výsledků jak pořadí, tak body. Výsledky je možné zveřejnit až po určení pořadí a bodů.

4.2.6 Uživatelé a oprávnění

Současný systém eviduje uživatele z několika zdrojů. Někteří jsou členy Rady a zastávají některou funkci. Jiní jsou pořadatelé závodu, respektive zástupci pořadatelského sboru. Ostatní jsou běžní, zaregistrovaní uživatelé, v systému nazýváni čtenáři. Celkem jsou používány tři kategorie uživatelských oprávnění – čtenář, autor a správce. Přihlašování čtenářů je rozdílné od přihlašování autorů.

Čtenář může pracovat s diskuzemi (vytvářet nové, přidávat komentáře), což ho zároveň opravňuje k přihlašování na závody. Autor má možnost přidávat články a správce spravuje všechny části systému.

V novém systému zůstane zachovaný obdobný model uživatelských oprávnění, upravený podle nového návrhu aplikace. Přihlášený uživatel má možnost přispívat do diskuzí a rezervovat startovní pořadí. Aby mohl zarezervovat i nový tým, který se Ligy ještě neúčastnil, bude mu umožněno vytvořit ho. Autor navíc může spravovat fotogalerie a články. Správce má přístup ke všem nastavením. V systému se operuje ještě s dynamickým oprávněním pořadatele. Tou je kontaktní osoba závodu a má právo měnit informace o závodu a sestavovat startovní pořadí družstev.



Obrázek 4.3: Use-case diagram uživatelských oprávnění.

4.2.7 Další možnosti přihlášení

Další novinkou v možnostech přihlašování uživatelů je možnost přihlášení se pomocí svého účtu na serveru Facebook prostřednictvím služby Facebook Connect. [12]

Mnoho lidí vlastní účet na tomto serveru, kde jsou permanentně přihlášení. API Facebooku poskytuje možnost využití přihlášení na tomto serveru i jiným službám. Na serveru

Facebook se vytvoří řídicí¹ aplikace, které přihlašovaný uživatel poskytne povolení pro používání jeho účtu. Zároveň nabízí bezpečnostní a identifikační klíče, pomocí kterých jsou identifikovány stránky, ke kterým se chce uživatel přihlásit.

Přihlášení probíhá tak, že uživatel klepne na tlačítko *Connect with Facebook*, které mu ukáže stránku s požadavkem na povolení přístupu řídicí aplikace k jeho účtu. Systém obdrží informaci o přihlášení uživatele a jeho uživatelské ID, pomocí kterého se dá spárovat účet na Facebooku a účet v hostitelském systému. Při prvním přihlášení se uživateli obvykle nabídne registrace (bez nutnosti zadat přihlašovací heslo). Při dalším přihlášení se zkontroluje, zda pomocí Facebook API ověřený uživatel je již registrován a automaticky se přihlásí.



The image shows a login interface. It has a 'Login' label next to a text input field containing 'Milan'. Below it is a 'Heslo' label next to an empty password input field. To the right of the password field is a button labeled 'Přihlásit se'. Below the password field is a blue link 'Zapomenuté jméno/heslo'. Below that is a green plus icon followed by a blue link 'Registrovat'. At the bottom is a blue button with the Facebook 'f' logo and the text 'Connect with Facebook'.

Obrázek 4.4: Přihlašovací formulář s možností přihlášení přes Facebook Connect.

Tento postup je standardní a byl využit i v tomto systému.

Přihlašování na cizí servery pomocí účtů svých uživatelů dnes nabízí více serverů.² Roli univerzálního účtu by také měla plnit služba OpenID, [17] jehož podporu nabízí pro své uživatele i Seznam.cz. [23] Díky jednoduchému přihlašování a uživatelské oblíbenosti byl zvolen pouze Facebook Connect. Od zavedení této služby si slibuji větší ochotu registrování uživatelů a jejich větší interakci se systémem.

4.2.8 Správa systému

V zájmu jednoduchého používání jsem se rozhodl nevytvářet plnohodnotnou administrační sekci, která by zaštiťovala veškerou správu systému, ale začlenil jsem správu dílčích částí přímo do systému. Přihlášení uživatelů je stejné jak pro běžné uživatele, autory i správce. Po přihlášení se uživatel dostane na stejnou stránku systému, ze které se registroval. Pouze se mu zamění obsah podle úrovně jeho oprávnění. Stránky, které přímo vyžadují, aby byl uživatel přihlášen, si při nesplnění této skutečnosti přihlášení vyžadají.

Některé stránky, jako jsou například články, závody nebo družstva nabídnou uživateli s dostatečným oprávněním odkaz na editační stránku. U jiných částí (například přihlašování na závody) se zamění obsah z tabulkového zobrazení na formulářové zobrazení, které přímo edituje danou část. Přihlášený uživatel tedy prochází systém jako běžný uživatel, ale nabízí se mu možnosti správy obsahu.

¹Pojem řídicí jsem zvolil pouze pro to, abych se na tuto aplikaci mohl dále odkazovat v textu. Oficiální název aplikace jsem nenalezl.

²Například Twitter, viz [20]

4.2.9 Automatická kontrola obsahu

Jak již bylo zmíněno, častým problémem je, že někteří uživatelé musí dělat práci za jiné uživatele. Abych kontrolu systému zautomatizoval, vytvořil jsem v administrační sekci stránku, která automaticky zkontroluje, zda jsou vytvořeny závody, uvedeny kontaktní údaje, připravené startovní pořadí pro závody a uvedeny výsledky k již proběhlým závodům. Uživatele graficky upozorní na nedostatky a nabídne ihned jejich nápravu.

Další možností kontroly může být upozornění správců emailem při překročení určité hraniční doby, např. tři týdny před závodem upozornit na nemožnost rezervace startovního pořadí nebo tři dny po závodě upozornit na stále nezadané výsledky. V odevzdávané verzi systému není tato funkčnost implementována.

Kontrola

Pro letošní rok byl vytvořen ročník.

Pro letošní ročník nejsou přiřazeny žádné závody. [Přidat nový závod](#)

Obrázek 4.5: Ukázka výstupu kontroly obsahu systému.

4.2.10 Statistiky

Přechod na nové databázové schéma umožňuje snadno získat a seskupit všechna potřebná data. Vést statistiky sportovních výsledků se přímo nabízí. Současné statistiky nejrychlejších časů a vítězů jednotlivých ročníků byly rozšířeny o různé statistiky dosahovaných časů a bodů. Všechny statistiky jsou zahrnuté do jedné složky. Přehled průměrných dosahovaných časů je evidován u každého družstva. Kromě zobrazení pomocí tabulky jsou některé vhodné statistiky zobrazeny formou grafu.

Kapitola 5

Analýza požadavků na zpracování grafů

Jedním z úkolu práce je vytvoření grafů pomocí moderních zobrazovacích technologií. Součástí Silverlight Toolkitu¹ je komponenta Chartig zajišťující vlastní vykreslení grafů. Z návrhu tedy odpadá grafický návrh a výpočet atributů grafu z datových řad. Inicializované komponentě se uvede zdroj dat a další vlastnosti, jako název grafu, typ a názvy datových řad.

5.1 Určení dat vhodných pro reprezentaci pomocí grafů

V systému byly nalezeny takové stránky, u kterých dává zobrazení pomocí grafů smysl. Ohled byl brán na zobrazení různých typů grafů. Zároveň bylo vhodné vybrat taková data, která jsou počtem svých hodnot vhodná pro grafové zobrazení. Jsou to:

- vítězové jednotlivých ročníků – koláčový graf²,
- průměrné časy sezón jednotlivých družstev – spojnicový graf,
- stav bodové tabulky ročníku – pruhový graf.

5.1.1 Koláčový graf

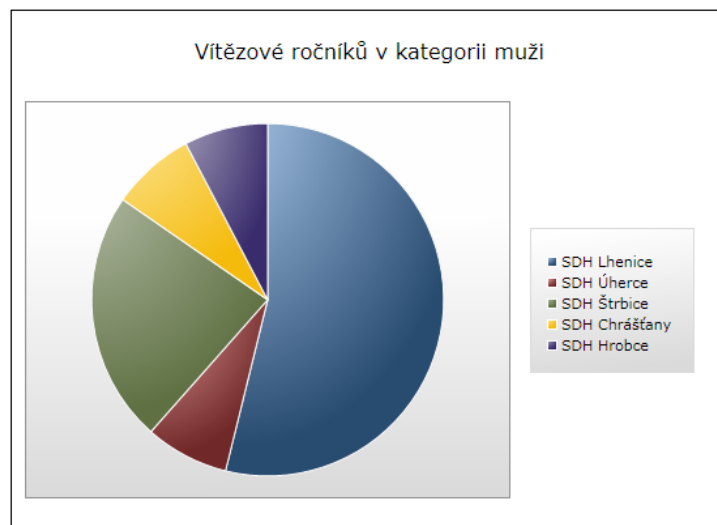
Informační systém umožňuje zobrazit výherce jednotlivých ročníků. Výherci se často opakují a zobrazení pomocí koláčového grafu dobře vyjádří poměr výher družstev vůči ostatním výhercům.

5.1.2 Spojnicový graf

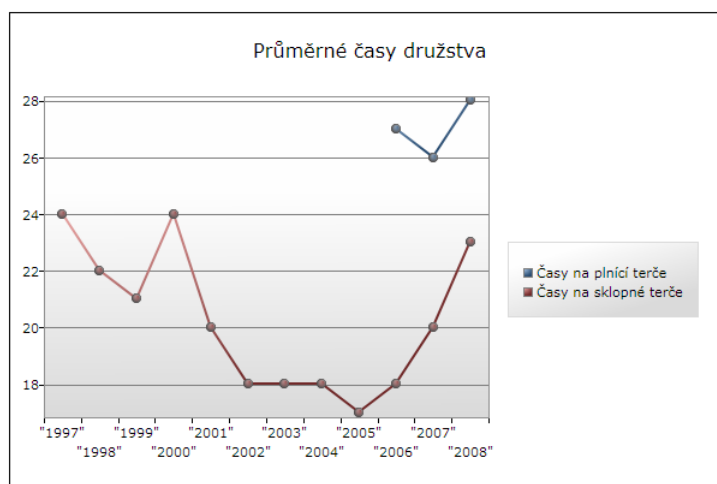
Vhodným ukazatelem kvality Ligy jsou průměrné časy jednotlivých sezón v jednotlivých kategoriích na jednotlivé terče. Tento výpis je vhodné zobrazit pomocí spojnicového grafu. Graf ukáže lepší se nebo horšící se tendenci Ligy.

¹Důvody vybrání této technologie jsou shrnuty v kapitole 3.

²Názvy typů grafů jsem volil podle zažitých konvencí z prostředí Microsoft Office a překladu použitého v [4].



Obrázek 5.1: Příklad koláčového grafu.



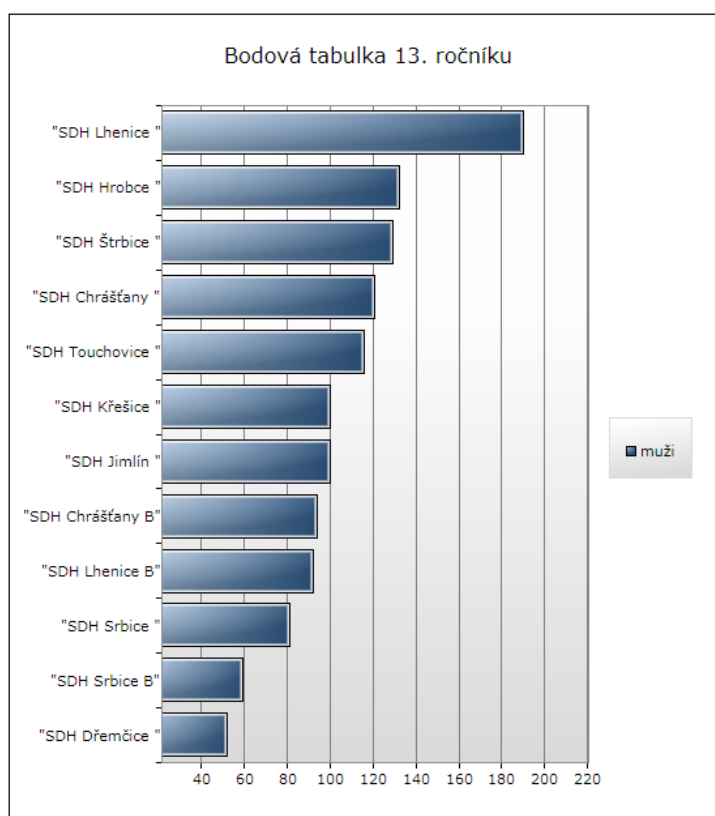
Obrázek 5.2: Příklad spojnicového grafu.

5.1.3 Pruhový graf

Mezi statistikami je zobrazen průběžný stav celkového bodování družstev. Družstva jsou řazena podle počtu dosažených bodů. Tabulka hůře vyjádří rozdíly mezi jednotlivými družstvy. Grafické zobrazení pomocí sloupcového grafu vhodně tyto rozdíly ukáže jako velikost „shodu“ mezi dvěma družstvy.

V loňské sezóně jsme takovouto statistiku pro námi pořádaný závod vytvořili. Graf ukázal velký náskok prvního týmu a poté tři různé skupiny podobně kvalitních týmů. Soutěžícím se grafická statistika líbila, dobře odhadli, na jaké umístění si ještě mohou dělat ambice.

Obdobným typem grafu je sloupcový graf (typ Column). Jedná se o tentýž graf, pouze otočený o 90 stupňů.



Obrázek 5.3: Příklad pruhového grafu.

5.2 Formát dat

Možnost použití grafů by neměla být vázána pouze na jeden systém, ale měla by být použitelná s jakýmkoli systémem. Zdroj dat by měl být nezávislý na souboru vykreslující grafy. Ideálně by měl být volen pomocí parametru.

Formát dat by měl být takový, aby bylo možné do něj data snadno exportovat i zpracovávat v grafové aplikaci. Takové možnosti přenosu dat poskytuje XML nebo JSON. S oběma formáty umí dobře pracovat jak skriptovací jazyky, tak Silverlight. Pro potřeby aplikace jsou

vlastnosti obou technologií shodné, s ohledem na menší datový objem byl vybrán JSON.

Formát byl zvolen tak, aby umožňoval vykreslení více grafů najednou a více datových řad různých typů v jednom grafu. Hlavním prvkem hierarchie je pole objektů, které popisují vlastní grafy. Každý objekt grafu obsahuje atributy grafu (název, rozměry) a popis datových řad. Ten je tvořen polem obsahujícím objekty datových řad. Ty dále obsahují atributy datové řady (její název) a vlastní hodnoty datové řady.

```
[
  // graf 1
  {
    'nazev' : 'Nejlepší časy sezóny 2010',
    'vyska' : 400,
    'sirka' : 500,
    'rady' :
    [
      // řada 1
      {
        'nazev' : 'Kategorie muži',
        'hodnoty' :
        'typ' : 'Line',
        [
          {'nazev' : 'SDH Lhenice', 'hodnota' : 17.34},
          ...
        ]
      },
      // řada 2
      {
        ...
      }
    ]
  },
  //graf 2
  {
    ...
  }
]
```

5.3 Integrace do systému

Grafová aplikace se do HTML stránky vkládá jako běžná Silverlight aplikace pomocí elementu `object`³. [13] Pomocí parametru `source` se určí cesta k vygenerovanému souboru XAP, obsahující grafovou aplikaci. Pomocí parametru `initParams` se určí atribut `dataUrl`, který odkazuje na zdroj dat. Pomocí dalších atributů elementu `object` a elementů `param` se nastaví povinné a volitelné vlastnosti vkládané aplikace.

³ *World Wide Web Consortium (W3C)* [online]. 2010 [cit. 2010-05-08]. Objects, Images, and Applets in HTML documents. Dostupné z WWW: <http://www.w3.org/TR/REC-html40/struct/objects.html#h-13.3>.

Kapitola 6

Použité technologie

6.1 Pro systém

Především serverové technologie jsou dány možnostmi používaného hostingu. Vzhledem k ceně i uživatelské oblíbenosti je dnes nejrozšířenější podpora trojice webového serveru Apache, skriptovacího jazyka PHP a databázového systému MySQL. Mnohdy je vyčítána především PHP a MySQL nevypěstlost v porovnání s konkurenčními technologiemi. S novými verzemi se podpora různých vlastností (např. OOP u PHP, databázové triggeru u MySQL) lepší.

6.1.1 PHP

Skriptovací jazyk PHP pochází z roku 1995, kdy se ze svých skromných začátků několika skriptů napsaných v jazyce PERL vyvinul až v objektový pokročilý nástroj splňující většinu požadavků pro vývoj serverových aplikací. [25] Podobně jako ostatní technologie související s tvorbou systémů ve webovém prostředí (HTML, MySQL), byl dlouho nabízen ve verzi 4, která, ačkoli byla obsáhlá, postrádala mnoho rysů moderních programovacích jazyků (podpora lepšího objektového návrhu, jmenné prostory, anonymní funkce apod.). Proto byla verze 5 vítanou, zejména díky lepší podpoře právě objektového programování. Nyní se jazyk nachází ve verzi 5.3, do které přibyla například podpora jmenných prostorů a je znát očištění jazyka od kontroverzních funkcí nebo funkcí nabízející rychlejší alternativy. [24]

6.1.2 JSON

JSON (JavaScript Object Notation) je formát dat, který je díky své jednoduchosti a univerzálnosti „opravdu ideálním jazykem“. [27] JSON má textový formát snadno zapsatelný i strojově analyzovatelný vycházející z konvencí jazyka C. Jazyk JSON pro přenos dat je využit pro předávání dat grafům a použitý Nette Framework ho využívá pro přenos dat v AJAXových požadavcích.

6.1.3 AJAX

AJAX je sloučení několika technik, které v celku přináší uživateli možnost lepší interakce s webovými aplikacemi. Termín AJAX označuje asynchronní JavaScript a XML. [9] Ve skutečnosti nemusí být dodržena ani jedna část, požadavek na server může být synchronní (čímž zaniká kouzlo použití AJAXu), namísto JavaScriptu se může použít jiný skriptovací jazyk a namísto XML se může použít JSON, prostý text nebo HTML.

Nic z toho ale nemění podstatu věci, a tou je vytvoření HTTP požadavku na pozadí stránky a obslužení odpovědi ze serveru pomocí skriptovacího jazyka na straně klienta. Server může na požadavek odpovědět jakkoli, nejčastěji novým obsahem, kterým aktualizuje stávající, požadavkem na přesměrování, popřípadě chybou či prázdnou odpovědí. Při správném návrhu aplikace se uživateli dostane komfortu v podobě obsluhování aplikace včetně změny obsahu bez nutnosti načítání celé stránky.

S tím souvisí i nevýhody použití. Webové aplikace se nejčastěji navrhuji jako graf, kdy jednotlivými stavy jsou generované stránky. Uživatelé jsou na tento systém zvyklí a procházení stavového prostoru jim usnadňují (kromě hypertextových odkazů) tlačítka zpět a vpřed. Aplikace využívající AJAX by s tímto měla počítat a implementovat ho na takové funkce, které přechod zpět (respektive vpřed) nevyžadují, protože změna na stránce obslužená AJAXem se standardně neuloží do historie prohlížeče.

Prohlížeče svou historii navštívených adres identifikují pomocí URI adres a ta se při použití AJAXu nemění (kdyby se měnila, došlo by k standardnímu HTTP požadavku). Nicméně součástí URI adresy je část označovaná podle RFC jako fragment, [2] v standardu HTML se používá název kotva. [15] Tato část je od adresy oddělena mřížkou a její hodnota identifikuje pozici v dokumentu. Na server se tato část pomocí HTTP nepřenáší. Pokud v těle HTML stránky není na tuto kotvu návaznost, interakce s takovýmto odkazem se nijak neprojeví.

Toho se dá využít právě pro ukládání AJAXových požadavků do historie prohlížeče. Při vyvolání takového požadavku se doplní na místo kotvy odkaz HTTP požadavku, popřípadě jinak v systému jednoznačně identifikovaný požadavek na server, což má za následek, že prohlížeč si tuto změnu uloží do historie, ale se samotným zobrazením stránky nic neprovede. Tu patřičným způsobem změni až obsluha zároveň vygenerovaného HTTP požadavku v pozadí. Obsluha klientské části AJAXu by poté měla při načtení stránky (událost *onload*) kontrolovat, zda se v kotvě URI adresy nenachází právě požadavek na obslužení pomocí AJAXového volání a ten provést. Buď zavolat obslužnou rutinu stejně jako při vyvolání požadavku uživatelskou interakcí¹, nebo přesměrovat na novou URL podle zadaných parametrů². Tím se zajistí načtení dodatečného obsahu při zadání takového URI do prohlížeče. Stránka není sice finálně načtena ihned po skončení načítání a její obsah se po chvíli opět změní, ale o tomto je uživatel z pravidla informován ukazatelem, že probíhá obsluha jeho požadavku na pozadí.

První z uvedených dvou řešení řešení jsem implementoval i ve vyvíjeném systému. V něm jsou AJAXové požadavky identifikovány specifickými parametry QUERY části URI schématu, které jsou ukládány na místo kotvy.

Stále tím není ošetřen případ, kdy uživatel klepnul na tlačítko zpět. Jednou z možností, jak reagovat i na tuto změnu je periodické hlídání změny kotvy v URI adrese³. Při využití tohoto řešení je nutné hlídat zdvojení požadavků na server. Jeden je vyvolán klepnutím na odkaz, druhý zjištěním změny v URI adrese.

6.1.4 Nette framework

Rutinní problémy při programování aplikací pomáhají řešit aplikační frameworky. Ty poskytují funkcionalitu, která není v samotném jazyku zahrnuta a přidávají další, specifickou pro účel jejich použití. Pro návrh aplikace jsem použil Nette Framework. Jedná se o malý,

¹Takto funguje například server Mapy.cz

²Takto funguje například server Facebook.com

³Toto řešení je nabízeno v rámci Mozilla Developer Center. [10]

ale rychlý [3] framework, jenž je určen pro tvorbu webů pomocí návrhového vzoru Model-View-Controller (MVC).

Tento návrhový vzor odděluje data do samostatných modelů, která jsou zobrazována pomocí pohledů – view. Mezičlánkem, který potřebná data od modelů přebírá a předává je pohledům a tvoří tak logiku aplikace, je kontrolní vrstva – controller.

Nette Framework dále poskytuje rozhraní pro tvorbu a obsluhu formulářů, přihlašování uživatelů a správu uživatelských rolí a oprávnění. Usnadňuje používání AJAXu v aplikacích snadnou změnou statické aplikace v dynamickou. Nette Framework umožňuje zneplatnění (*invalidaci*) pouze části stránky (*snippetu*) a její odeslání do prohlížeče, namísto odeslání celé stránky. Aktualizovaná část se zamění pomocí obslužných Javascriptových rutin. Takto lze jedním požadavkem zaměnit libovolné množství snippetů během jednoho požadavku, pro každý stačí jeden příkaz v controlleru stránky.

Použití Nette Frameworku přineslo řadu úspor jak časových, tak pracovních. Nebylo potřeba programovat součásti nutné pro správný chod aplikace, stačilo se soustředit na vlastní logiku vytvářeného systému.

6.1.5 dibi framework

Tento framework přináší, mimo jiné, podporu pro postupné tvoření dotazů. Toho je využito v modelech, kde se dotazy pro nalezení jednoho záznamu z databáze oproti nalezení všech záznamů liší pouze klauzulí WHERE. Framework podporuje tedy znovupoužitelnost a eliminuje tím možnost vzniku chyb při úpravě dotazů. Dále ošetřuje všechny předané parametry na útok SQL injection. Předávané parametry se uvozují modifikátory (např. `%i`, podobně jako u knihovní funkce `printf` jazyka C) a ty jsou před vložením do dotazu podle svých datových typů ošetřeny správnou kódovací funkcí.

6.2 Pro grafy

6.2.1 XAML

XAML je značkový jazyk vycházející z XML. V aplikacích Silverlightu se využívá pro tvorbu aplikačních rozhraní. XAML, na rozdíl od konkurenčních technologií, je založen na textovém, nikoli binárním formátu při změně obsahu vytvoří nový XAML soubor, který obdrží klientská aplikace při další návštěvě stránky. [7]

6.2.2 C#

C# je vysokoúrovňový objektově orientovaný jazyk vyvinutý firmou Microsoft. [22]

6.2.3 Silverlight a Silverlight Toolkit

Silverlight je multiplatformní plug-in modul společnosti Microsoft rozšiřující schopnosti prohlížeče o nové prezentační možnosti. Jedná se o implementaci .NET Frameworku pro tvorbu bohatých interaktivních aplikací (RIA) ve webovém prostředí⁴. [21] Vývojářům umožňuje vytvářet komplexní aplikace fungující v každém podporovaném prohlížeči stejně. Uživatelům potom ucelenější a interaktivnější uživatelské prostředí.

⁴V poslední verzi 4.0 již aplikace může fungovat nezávisle na prohlížeči jako tzv. standalone aplikace. [1]

Silverlight Toolkit obohacuje Silverlight o nové komponenty a aplikace, které se snadno začlení do aplikace a pomohou vytvořit celistvý produkt. Kódy aplikace jsou uvolněny pod open-source licencí.

Kapitola 7

Implementace systému

7.1 Struktura aplikace

Používaný Nette Framework podporuje tvorbu pomocí MVC návrhového vzoru. Soubory očekává v doporučené adresářové struktuře. Ve složce `/app/` jsou umístěny skripty aplikace a šablony stránek, ve složce `/document_root/` soubory přístupné pomocí prohlížeče (index.php, kaskadové styly, obrázky, soubory s JavaScriptem), ve složce `/libs/` knihovny nebo na aplikaci nezávislé skripty.

Pomocí modulu `mod_rewrite` webového serveru Apache jsou všechny požadavky na html a php soubory přeměrovány na soubor `index.php`, ve kterém jsou nastavené konstanty obsahující cesty podle výše popsané struktury a je spuštěna aplikace hlavním souborem `/app/bootstrap.php`. Takovýto přístup zvyšuje zabezpečení aplikace, protože přístup přes webový prohlížeč je možný pouze do adresáře `/document_root/`.

Samotná aplikace se dělí na tři složky. Složku `/app/models/` obsahující soubory s modely, složku `/app/presenters/` obsahující kontrolery, kterým se v kontextu Frameworku říká presentery a složku `/app/templates/` obsahující šablony stránek.

Páry pohled-presenter často kopírují jednotlivé entity ER modelu, a ty obsahují i samostatný model. Některé jsou pouze virtuální, například statistiky mají jak pohledy a presenter, ale vlastní model již ne. Data přebírají z modelu výsledků.

7.2 Využití komponent pro vložení grafu do stránky

Specifikem Nette Frameworku je početné využívání *komponent*. Komponenta je určitý logický blok stránky (například anketa) mající vlastní logiku a šablony. Komponenty jsou často nezávislé na konkrétní aplikaci. Výhodou použití komponent je kromě zmíněné možnosti znovupoužití v jiné aplikaci, jejich snadné vykreslení vícekrát v rámci jedné stránky a jejich samostatná funkčnost. Komponenta lze, stejně jako snippet, snadno zneplatnit a tím si ji nechat pomoci AJAXu překreslit. Výhodou komponenty je i její snadné vložení do šablony stránky pomocí připraveného makra `{widget názevKomponenty}`.

Z těchto důvodů jsem se rozhodl kód nutný pro vykreslení grafů umístit právě do komponenty. Na určené místo v šabloně stránky se vloží `{widget grafy}` doplněné parametry aktuálního grafu. Tím jsou rozměry grafu a cesta k datům grafu. Výsledný zápis tedy vypadá `{widget grafy array(450, 620, 'Statistiky:grafVitezoveRocniku')}`, kde zápis `array(...)` udává pole parametrů, které se předá instanci třídy komponenty. Řetězec `'Statistiky:grafVitezoveRocniku'` udává odkaz na funkci presenteru, která vrátí potřebná data.

Zde zadané rozměry jsou pouze předpokládáné, komponenta si stáhne data pro graf a podle zadaných rozměrů jednotlivých grafů určí správné rozměry.

7.3 Správa systému

Jak bylo již uvedeno, nebyla vytvořena zvláštní sekce, která by obsahovala správu systému. Přihlášeným uživatelům se obsah stránky náležitě změní, aby ji mohl spravovat podle úrovně svého oprávnění.

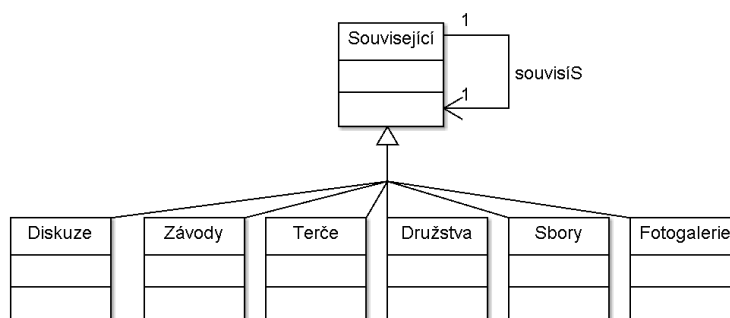
I tak existují určité databázové tabulky, které musí být spravovány uživatelem, ale v systému nemají svůj pohled. Jedná se především o různé číselníky, obsahující například typy terčů, sportovní kategorie, ale i bodové hodnocení výkonů a stránky, u kterých je veškerý obsah tvořený uživateli a nevlastní svůj vlastní presenter. Stejně tak má systém určité funkce, které by se daly označit za ryze administrativní. Pro všechny tyto části nebyla vytvářena zvláštní sekce, pouze stránka, obsahující odkazy na všechny činnosti.

Zároveň všechny zmíněné úkony vyžadují, aby byl uživatel přihlášen. Bázový presenter obsahující různá nastavení systému, inicializace komponent a společné metody byl poděděn do nového presenteru pojmenovaného **SecuredPresenter**, který pro vykreslení svých pohledů vyžaduje přihlášení uživatele. Z toho presenteru vychází právě presentery provádějící výše popsané činnosti.

7.4 Možnosti diskuzí a souvisejících položek

Možnosti diskuzí byly značně rozšířeny. Komentáře lze nyní přidávat nejen k obecným tématům, ale i k družstvům, sborům, závodům, terčům a článkům. Bylo nutné vyřešit nepovinnou závislost témat na vyjmenovaných entitách. Především z pohledu uživatelského rozhraní a databázového schématu. Jednak je nutné při vytváření tématu sdělit, ke které entitě se téma bude vztahovat, také při zakládání diskuze musí uživatel zvolit, na který prvek entity uživatel reaguje.

V důsledku se jedná o převod generalizace v ER diagramu na schéma relační databáze. Vzhledem k disjunktnosti entit je vhodným řešením převést každou entitu na svojí vlastní tabulku a nadřazenou entitu modelovat jako vztah v relační databázi pomocí dvou atributů v tabulce ukazující na závislou tabulku. Jeden atribut udává název závislé tabulky, druhý primární klíč této tabulky.



Obrázek 7.1: ER diagram části obstarávající vztahy souvisejících položek.

Diskuze ale nejsou jediné, které se mohou k něčemu vztahovat. Bylo vhodné, aby i fotogalerie a články mohly následovat související téma. Z těchto důvodů jsem se rozhodl

související vztahy ukládat ve vazební tabulce, která obsahuje názvy souvisejících tabulek a jejich primární klíče identifikující položku. Aby bylo možné využít indexy tabulek pro snadnější vyhledání (zároveň kontrolování duplicitních záznamů pomocí unikátního indexu) je jedna tabulka uvedena jako primární a druhá sekundární. Závislosti obou položek jsou vzájemné, proto se musí uložit závislost položek A a B tak, kdy jednou je primární položka A, podruhé položka B.

Vytvoření obecných závislostí umožňuje vytvářet libovolné dvojce z předem dané množiny tabulek, které pro to mají připravenou aplikační podporu. Tou je možnost zobrazení dat z tabulky ve formulářovém poli select a naopak zobrazení odkazu na související položku v šabloně.

Fotogalerie: Soutěž v Křešicích

Související

- Družstvo: [SDH Křešice muži](#)
- Družstvo: [SDH Duchcov muži B](#)
- Družstvo: [SDH Boletice nad Labem muži](#)
- Závod: [Duchcov](#), v sobotu 11. října 1997 v 10.00
- Závod: [Vejprty](#), v sobotu 14. září 2002 v 10.00

Obrázek 7.2: Ukázka výpisu souvisejících položek.

Druhou možností by bylo neuvádět vztah mezi položkami jako primární a sekundární. V tomto případě by dotaz pro nalezení záznamu musel vyhledávat jak z první dvojice název tabulky-primární klíč, ale také z druhé. Při plánované četnosti v řádech několika málo desítek souvisejících záznamů ročně se může toto řešení jevit z hlediska využití paměti (dat databáze i indexů) lepší a rychlostně srovnatelné. Z tabulky se bude převážně číst, proto byla zvolena varianta využívající indexy a zdvojené vkládání řádků. Vyhledávaná data bude schopen SŘBD použít přímo z indexu, což ulehčí jejich časté hledání. Zároveň v dnešní době není s datovou náročností v takovýchto malých měřítcích problém¹. Zdvojené vložení řádků obslouží databázový příkaz uložený v modelu, tudíž transparentně pro logiku aplikace.

V editačním formuláři položky podporující určování související položek je uživateli nabídnuto prostřednictvím prvku select vybrání související tabulky a podle toho druhý prvek select obsahující přehled položek tabulky.

Možnost založit novou diskuzi má uživatel samozřejmě také ze stránky se související položkou. V tom případě je volba související položky irelevantní a určí se automaticky.

7.5 Fotogalerie

Podobný princip určení souvisejících položek jsem použil i v podsystému fotogalerií. Při vytváření fotogalerie uživatel může zvolit související tabulku a podle ní vybrat související položku. Ostatní funkce z předešlého systému zůstaly zachovány.

¹Databáze po převodu všech závodů a výsledků má přibližně 300 kB, místo pro databázi nabízené poskytovatelem webového prostoru se počítá na desítky MB.

7.6 Statistiky

System nově eviduje daleko více statistik, než původní systém. Kromě sportovních statistik (nejrychlejší časy, nejlepší družstva, ...) jsou nabízeny pořadatelské statistiky (počty pořádaných závodů).

Sportovní výsledky obsahují většinou různé průměrné časy. Družstva většinou podávají po celý rok přibližně stejné výsledky, občas se stane, že se pokus nepodaří zaběhnout podle představ a čas je výrazněji horší, než průměr. Při implementaci tedy proběhlo srovnání průměrných časů a středních hodnot časů. Mnohdy se střední hodnoty lišily i o několik sekund oproti průměrným časům. Jak k horšímu, spíše však k lepšímu. Aby průměrné časy nebyly ovlivňovány takovýmito výchyly, jsou všechny počítány jako střední hodnoty. Pouze pokud jsou k dispozici jen dva časy, je určen jejich průměr.

Kapitola 8

Implementace grafů

8.1 Úvod

Uživatelské rozhraní aplikace se v Silverlight popisuje pomocí jazyka XAML v souborech s příponou xaml. Aplikační kód je možné psát pomocí jazyka C# nebo Visual Basic.net. Soubory jazyka C# mají příponu cs, soubory Visual Basicu mají příponu vs. V tomto projektu je používán pro logiku aplikace jazyk C#.

Silverlight Toolkit rozšiřuje možnosti Silverlightu o definici grafů jak pomocí značek XAML, tak pomocí aplikačního rozhraní ostatních podporovaných jazyků.

8.2 Návrh implementace

Aby bylo možné s grafy a projektu pracovat, je nutné připojit do jmenného prostoru XAML stránky odkaz na knihovnu s grafy. Programování uživatelského prostředí probíhá pomocí značkování v souboru s uživatelským rozhraním. Pomocí značky `Chart` se vytváří nový graf. Ke grafu je možné připojit více datových řad najednou vně vnořené značky `Chart.Series`. Každá datová řada je určena značkou, jejíž název je odvozen od typu datové řady. Spojnicový graf je určen značkou `LineSeries`, sloupcový graf značkou `ColumnSeries`, koláčový pomocí `PieSeries`, pruhový značkou `BarSeries`. Silverlight Toolkit nabízí podporu ještě pro další čtyři druhy grafů, které nebyly implementovány. Datovým řadám se nastaví vhodný zdroj dat, který je popsán dále v textu. O vlastní vykreslení se postará aplikace sama.

Tímto způsobem je možné definovat předem daný počet grafů. Jedním z požadavků aplikace je možnost vykreslení více grafů v rámci jedné instance. Z těchto důvodů bylo na XAML stránce definováno pouze vertikální rozložení pomocí layoutu a veškeré grafy jsou tvořeny pomocí C# aplikačního kódu na vyžádání podle definicí popsané vstupními daty.

8.3 Moje implementace

Uživatelské prostředí je popsáno v souboru `Grafy.xaml` a logika v souboru `Grafy.xaml.cs`. Hlavním souborem aplikace, který je inicializován po načtení v prohlížeči, je soubor `App.xaml.cs`.

Start aplikace obstarává třída `App` odvozená od třídy `Application`, která obstarává rutiny prováděné po startu, při chybě a před ukončením aplikace.

Konstruktor třídy nastaví obslužné funkce, ve kterých se spustí logika vlastní aplikace umístěná ve třídě `MainPage` v souboru `MainPage.xaml.cs`. Mezi parametry standardně

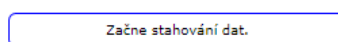
předávané mezi funkcemi jsem přidal i objekt `StartupEventArgs` obsahující startovní parametry uvedené v místě vložení komponenty v těle HTML stránky v elementu `param` se jménem `InitialParams`.

8.3.1 Stažení dat

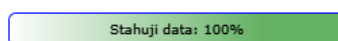
Po inicializaci stránky s uživatelským rozhraním se stáhnou uživatelem zadaná data určená parametrem `dataUrl`. Stažení dat obstarává knihovní třída `Webclient`. Spustí se asynchronní požadavek, po jehož dokončení se zavolá funkce zpracovávající data.

Třída `WebClient` je zjednodušenou nástavbou nad třídou `HttpWebRequest`. Třída `HttpWebRequest` umožňuje vytváření vlastního HTTP požadavku včetně tvorby vlastních hlaviček požadavku. Třída `WebClient` poskytuje základní funkčnost HTTP protokolu (pouze práce s metodami GET a PUT) a omezenější repertoár funkcí. [8, str. 188] Nicméně pro potřeby jednoduchého stažení dat je dostačující.

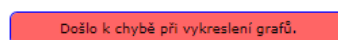
Před stažením dat a vykreslením grafů se v plátně aplikace ukáže ukazatel [8, str. 203], který uživateli dává najevo svou činnost prostřednictvím textové zprávy a grafického znázornění průběhu stahování dat. Po inicializaci se zobrazí zpráva „Začne stahování dat.“ Třída `WebClient` poskytuje funkcionalitu, pomocí které můžeme zjistit dosavadní průběh stahování. Na tuto funkcionalitu napojená funkce mění stav ukazatele - vypisuje, kolik procent dat bylo staženo a zeleným pruhem měnícím svoji délku podle počtu procent tuto skutečnost naznačuje graficky. Po správném stažení dat se ukazatel skryje, při chybě se ukazatel podbarví červeně a zobrazí zprávu „Došlo k chybě při získání dat pro graf.“ Stejně tak při jakékoli chybě při parsování dat se zobrazí upozornění.



Obrázek 8.1: Ukazatel průběhu stahování po inicializaci.



Obrázek 8.2: Ukazatel průběhu stahování po stažení všech dat.



Obrázek 8.3: Ukazatel průběhu stahování při chybě.

8.3.2 Zpracování dat

Po správném stažení jsou data popisující grafy a jejich datové řady uložena v paměti. Vstupním proudem dat se inicializuje objekt obsahující pole dat formátu JSON. Podle předepsaného formátu dat se nyní rozparsují v několika průchodech cykly procházející postupně jednotlivá pole objektů datového souboru formátu JSON. Pro každý graf se vytvoří nová instance třídy `Chart`, která se po nastavení všech datových řad a parametrů vloží do layoutu. Pro každý graf se projdou datové řady, jejichž hodnoty nezávislé i závislé osy

se připraví do seznamu hodnot aktuální datové řady. Podle typu datové řady se následně vytvoří instance požadovaného typu datové řady (linear pro spojnicový, bar pro pruhový, column pro sloupcový, pie pro koláčový). Datové řadě se nastaví titulek zadaný v datovém souboru a hodnoty datových os. Pouze u koláčového grafu není titulek za potřebí, protože v něm může být zobrazena pouze jedna datová řada.

Pokud je definováno více grafů do jedné instance aplikace, vytvoří se odstup mezi grafy.

Grafy se následně vykreslí v prohlížeči. Při najetí myši se uživateli zvýrazní aktuální bod a klepnutím si ho může označit.

Kapitola 9

Zhodnocení výsledků

9.1 Přínos systému pro Krušnohorskou ligu

Největším přínosem by bylo, kdyby systém splnil všechny požadavky a dočkal se reálného nasazení. Před dokončením technické zprávy byl systém zveřejněn v pilotním provozu a byl na něj uveden odkaz na původních stránkách Ligy. Setkal jsem se zpětnou odezvou, která byla pozitivní. Uživatelům se líbilo především přihlašování na závody a také „ta propojení jsou dost zajímavá“¹ Nedostatky starého systému se projevili týden před prvním závodem sezóny 2010, kdy pořadatelé závodu v Měrunicích opět neotevřeli možnost rezervace pořadí a uživatelé začali zmatkovat a zapisovat přihlášky do různých témat v diskuzi. Z těchto důvodů se zdají být provedené změny ku prospěchu věci. Po domluvě se zástupci Rady Ligy byl dohodnut termín předpokládaného spuštění nové verze na letošní prázdniny, kdy nejsou plánovány žádné závody a převod nenaruší možné problémy.

9.2 Přínos systému pro autora

Při vytváření systému jsem se snažil využít znalosti nabyté při dosavadním studiu. Jednalo se především o zkušenosti z oblasti uživatelských rozhraní, databázových systémů a informačních systémů. Vzhledem k tomu, že jsem implementoval systém s plánovaným dlouhodobým a četným provozem, byl jsem nucen důkladně se zamyslet nad všemi aspekty návrhu. Při tom vyvstalo vždy několik otázek optimálního návrhu a možných řešení, což mě přimělo k hlubšímu studiu problému.

9.3 Návrh dalších prací na systému

Systém v této fázi není připraven na reálný provoz. Kromě chybějícího vzhledu chybí systému jiné části, které jsou žádoucí.

Jednou z nich je anketní subsystém. Ankety jsou součástí starého systému a měly by být i součástí nového systému. Pro požadavky bakalářské práce nejsou ankety prioritní a tak budou doimplementovány dodatečně. Jejich začlenění nijak neovlivní současný návrh.

Stejně tak fotogalerie obsahují pouze základní podporu. Naprogramovány byly pouze části nutné pro naznačení budoucí verze, jakýsi prototyp. Fotogalerie jsou součástí subsystému souvisejících položek a chtěl jsem tím naznačit, jak je možné snadno přiřadit fotogalerii

¹ Příspěvek uživatelky Karenka v diskuzi KL: *Diskuze - Krušnohorská liga* [online]. 2005 [cit. 2010-05-04]. Dostupné z WWW: <http://krusnohorskaliga.cz/kforum.php?akce=topic&Top_ID=98>.

k článku nebo například závodu. Opět se jedná o novinku v systému a chtěl jsem získat zpětnou odezvu od uživatelů.

S fotografiemi okrajově souvisí nahrávání dokumentů. Jak soubory, tak fotografie sdílí stejnou databázovou tabulku. Návrh uživatelského rozhraní a implementačních detailů práce se soubory bude součástí další práce na systému.

Speciálním typem souborů, které ovšem nebude uživatel nahrávat, bude příprava pro pořadatele. Pořadatel má podle pravidel povinnost každému týmu při vyhlášení předat výsledky dané kategorie. Každý pořadatel si připraví tabulku (většinou v aplikaci Microsoft Office Excel), do které výsledky zadává. Provedení tabulky odpovídá znalostem každého pořadatele. Kancelářský balík Microsoft Office využívá od verze 2007 nový formát pro ukládání dat – Open XML. Ve stejném roce vznikl projekt umožňující vytváření tabulek aplikace Excel v PHP. [18] Nabízelo by se připravit tabulku pro pořadatele právě pomocí rozšíření PHPExcels. Součástí tabulky by byl bodový stav před závodem a názvy přihlášených týmů. Pořadatel by vyplňoval pouze výsledky k týmům. Výhodou by byl jednotný vzhled výsledných tabulek a ulehčení práce pořadatelů. Liga se sice nepotýká s nedostatkem pořadatelů závodů, ale kromě tradičních sborů by se přidaly i nové. Toto rozšíření by bylo otázkou vzdálenější budoucnosti, například příští sezóny.

Lepším řešením výše zmíněného rozšíření by bylo, kdyby pořadatel mohl přímo v průběhu závodu ukládat výsledky do systému. Po skončení závodů by se mu nabídla tisková sestava. Výsledky by potom nemusely být podruhé zadávány do systému. Nevýhodou je ovšem nutnost mít v místě konání závodu internetové připojení.

Co neuměl stávající systém a neumí ani nový je upozorňování uživatele na nové příspěvky ve sledovaných diskuzích. Oznamování by mohlo být řešeno zasíláním emailů z diskuzí, do nichž uživatel přispěl. Bylo by záhodno, aby se toto rozšíření dostalo do finální verze.

Literatura

- [1] Overview : The Official Microsoft Silverlight Site.
<http://www.silverlight.net/getstarted/overview.aspx>, 2010 [cit. 2010-05-04].
- [2] Berners-Lee, T.: RFC 3986 (rfc3986) - Uniform Resource Identifier (URI): Generic Syntax. <http://www.faqs.org/rfcs/rfc3986.html>, 2005.
- [3] Daněk, P.: Velký test PHP frameworků - Root.cz.
<http://www.root.cz/clanky/velkytestphpframeworku2008/>, 2008-08-21 [cit. 2010-05-02].
- [4] Gibilisco, S.: *Statistika bez předchozích znalostí*. Computer Press, 2009, iISBN 978-80-251-2465-9.
- [5] Gube, J.: 8 CSS Techniques for Charting Data [online].
http://sixrevisions.com/css/css_techniques_charting_data/, 2008-09-20 [cit. 2010-04-10].
- [6] Hassman, M.: HTML 4 5 6... <http://html456.blogspot.com/>.
- [7] Luboslav Lacko: *Silverlight 2.0 [online]*. Microsoft.
- [8] Papa, J.: *Silverlight – datové služby*. Zoner Press, 2009, iISBN 978-80-7413-041-0.
- [9] Resig, J.: *JavaScript a Ajax :moderní programování webových aplikací*. Computer Press, 2007, iISBN 978-80-251-1824-5.
- [10] Stenhouse, M.: Fixing the Back Button and Enabling Bookmarking for AJAX Apps.
[https://developer.mozilla.org/en/Fixing_the_Back_Button_and_Enabling_Bookmarking_for_AJAX_Apps_\(external\)](https://developer.mozilla.org/en/Fixing_the_Back_Button_and_Enabling_Bookmarking_for_AJAX_Apps_(external)), 2005-06-15 [cit. 2010-05-02].
- [11] WWW Stránky: The canvas element – HTML 5.
<http://dev.w3.org/html5/spec/the-canvas-element.html>.
- [12] WWW Stránky: Facebook Connect.
http://wiki.developers.facebook.com/index.php/Facebook_Connect.
- [13] WWW Stránky: How to: Add Silverlight to a Web Page by Using HTML.
<http://msdn.microsoft.com/en-us/library/cc189089%28VS.95%29.aspx>.
- [14] WWW stránky: Kronika KL. <http://krusnohorskakaliga.cz/kronika.php>.
- [15] WWW Stránky: Links in HTML documents.
<http://www.w3.org/TR/html401/struct/links.html#h-12.2.1>.

- [16] WWW Stránky: Microsoft Silverlight Toolkit.
<http://silverlight.codeplex.com/Wikipage>.
- [17] WWW Stránky: OpenID Foundation website. <http://openid.net/>.
- [18] WWW Stránky: PHP Excel. <http://phpexcel.codeplex.com/>.
- [19] WWW Stránky: Přehrávač videí YouTube HTML5.
<http://www.youtube.com/html5>.
- [20] WWW Stránky: Sign in with Twitter.
<http://apiwiki.twitter.com/Sign-in-with-Twitter>.
- [21] WWW Stránky: Silverlight Overview.
<http://msdn.microsoft.com/en-us/library/bb404700%28VS.95%29.aspx>.
- [22] WWW Stránky: Visual C#.
<http://msdn.microsoft.com/en-us/library/kx37x362%28VS.90%29.aspx>.
- [23] WWW Stránky: Otevřená registrace - Seznam blog [online].
<http://seznam.sblog.cz/2008/12/11/71>, 2008-12-11 [cit. 2010-04-15].
- [24] WWW Stránky: PHP: Deprecated features in PHP 5.3.x - Manual [online].
<http://cz.php.net/manual/en/migration53.deprecated.php>, 2010-04-30 [cit. 2010-05-02].
- [25] WWW Stránky: PHP: History of PHP - Manual [online].
<http://cz.php.net/manual/en/history.php.php>, 2010-04-30 [cit. 2010-05-02].
- [26] WWW Stránky: Rich Internet Application Market Share.
http://www.statowl.com/custom_ria_market_penetration.php, [cit. 2010-04-10].
- [27] WWW Stránky: Úvod do JSON [online]. <http://json.org/json-cz.html>, [cit. 2010-05-02].

Dodatek A

Obsah CD

Ve složce `/www/` je uložena webová aplikace. Ve složce `/grafy/` jsou uloženy zdrojové soubory ke grafům. Ve složce `/latex/` jsou uloženy zdrojové soubory k technické zprávě.

Návod pro zprovoznění aplikace je uvedený v kořenové složce v souboru `install.txt`.